Exemplo 4:

Seja:

tipo exemplo = matriz [1 : 3, 0 : 4] inteiro;

exemplo: VEJABEM:

	0	1	2	3	4
1					
2					
3					

VEJABEM

1 – Para percorrer a matriz linha por linha:

Fixe a Linha
Varie a Coluna

2 – Para percorrer a matriz coluna por coluna:

Fixe a Coluna
Varie a Coluna

```
1. I ← 1;
   enquanto I ≤ 3 faça
   |  J ← 0
   |  enquanto J ≤ 4 faça
   |  |     VEJABEM [I, J] ← I + J;
   |  |     J ← J + 1;
   |  fim enquanto;
   |  I ← I + 1;
   fim enquanto;

2. J ← 0
   enquanto J ≤ 4 faça
   |  I ← 1;
   |  enquanto I ≤ 3 faça
   |  |     VEJABEM [I, J] ← I + J;
   |  |     I ← I + 1;
   |  fim enquanto;
   |  J ← J + 1;
   fim enquanto;
```


ou então:

```
para J de 0 até 4 passo 1 faça
|  para I de 1 até 3 passo 1 faça
|  |     VEJABEM [I, J] ← I + J;
|  fim para;
fim para;
```

Exemplo 5:

Como no caso de vetores, também para matrizes podemos ter comandos concisos para inicialização, leitura e impressão.

Por exemplo, seja MAT definida por:

```
tipo m = matriz [1 : 10, 1 : 8] inteiro;
m: MAT;
```

O comando:

```
leia (MAT); {uma linha por cartão}
```

é equivalente ao seguinte trecho:

```
para I de 1 até 10 passo 1 faça
|  leia (para J de 1 até 8 passo 1 faça MAT [I, J] fim para);
fim para.
```

Do mesmo modo, podemos inicializar uma matriz com todos os seus elementos iguais a zero escrevendo:

MAT ← 0;

Para inicializar uma matriz com valores diferentes, usamos um comentário para indicar se o preenchimento deve ser feito por linha ou por coluna.

Por exemplo, seja MAT1 do tipo:

```
tipo m1 = matriz [1 : 2, 1 : 3] inteiro;
m1: MAT1
```

O comando:

```
MAT1 ← (10, 20, 30, 40, 50, 60); {por linha}
```

dá à matriz a seguinte configuração inicial: